TMS99650 Multiprocessor Interface (MPIF) Data Manual

Advanced Peripherals

TEXAS INSTRUMENTS

TMS99650
Multiprocessor Interface (MPIF)
Data Manual

TEXAS
INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments assumes no responsibility for infringement of patents or rights of others based on Texas Instruments applications assistance or product specifications, since TI does not possess full access to data concerning the use or applications of customer's products. TI also assumes no responsibility for customer product designs.

This document contains information on a new product. Specifications are subject to change without notice.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1 GENERAL DESCRIPTION

The TMS99650 Multiprocessor Interface device (MPIF) provides a bit-parallel, asynchronous communications interface for passing messages and data between two processors or processor systems. It represents a standard peripheral interface consisting of eight programmable registers at each of its two ports and furnishes access to 256 bytes of random-access memory (RAM) used to buffer data transmitted between ports. The MPIF supplies arbitration logic to resolve RAM-access conflicts between the two processor systems. The MPIF can be used to connect virtually any 8-bit or 16-bit microprocessor to any other 8-bit or 16-bit microprocessor having the capability of interfacing to standard memory or peripheral devices.

1.2 KEY FEATURES

- Two fully independent, asynchronous processor interfaces or ports
- Eight programmable 8-bit registers at each port
- Status and control registers
- Interrupt request control at each port
- Pointer registers that can be incremented following a RAM access
- Message registers to pass data between ports independent of RAM arbitration logic
- 256 bytes of RAM addressed indirectly using pointer registers
- Hardware support for utilization of RAM as first-in first-out (FIFO) buffer between ports
- Width of data path expandable in 8-bit increments
- Internal arbitration of asynchronous RAM-access conflicts
- Optional READY signal for concurrent use of the memory by both ports
- READY synchronized on chip; CLKIN required only if READY is used
- Hardware lockout capability provided to support test-and-set, test-and-clear operations
- Software lockout facility with interrupt for confirmation
- Single 5-V supply
- 40-pin DIP package
- N-channel silicon-gate technology

2. FUNCTIONAL DESCRIPTION

2.1 ARCHITECTURE

The Multiprocessor Interface (MPIF), shown in Figure 1-1 as a block diagram, is built around a 256 × 8-bit static random-access memory and includes two complete microprocessor interfaces and two complete data paths. Each data path connects the microprocessor interface to the appropriate on-chip registers under the control of the external interface control signals, register select lines, and the arbitration latch.

Both interfaces have access to the RAM via data registers, which are simple bidirectional buffers between the RAM and the data paths and involve no storage.

Each data register has associated with it an address pointer register that supplies the address to the RAM when the corresponding data register is used. The address registers can be written to and read from both microprocessor interfaces.

Two message registers are provided, one assigned to each port. Each interface can read and write its own message register but only read that of the other interface.

The control register provides for the configuration and control of the MPIF. It includes the enable bits for the various sources of interrupt request on-chip.

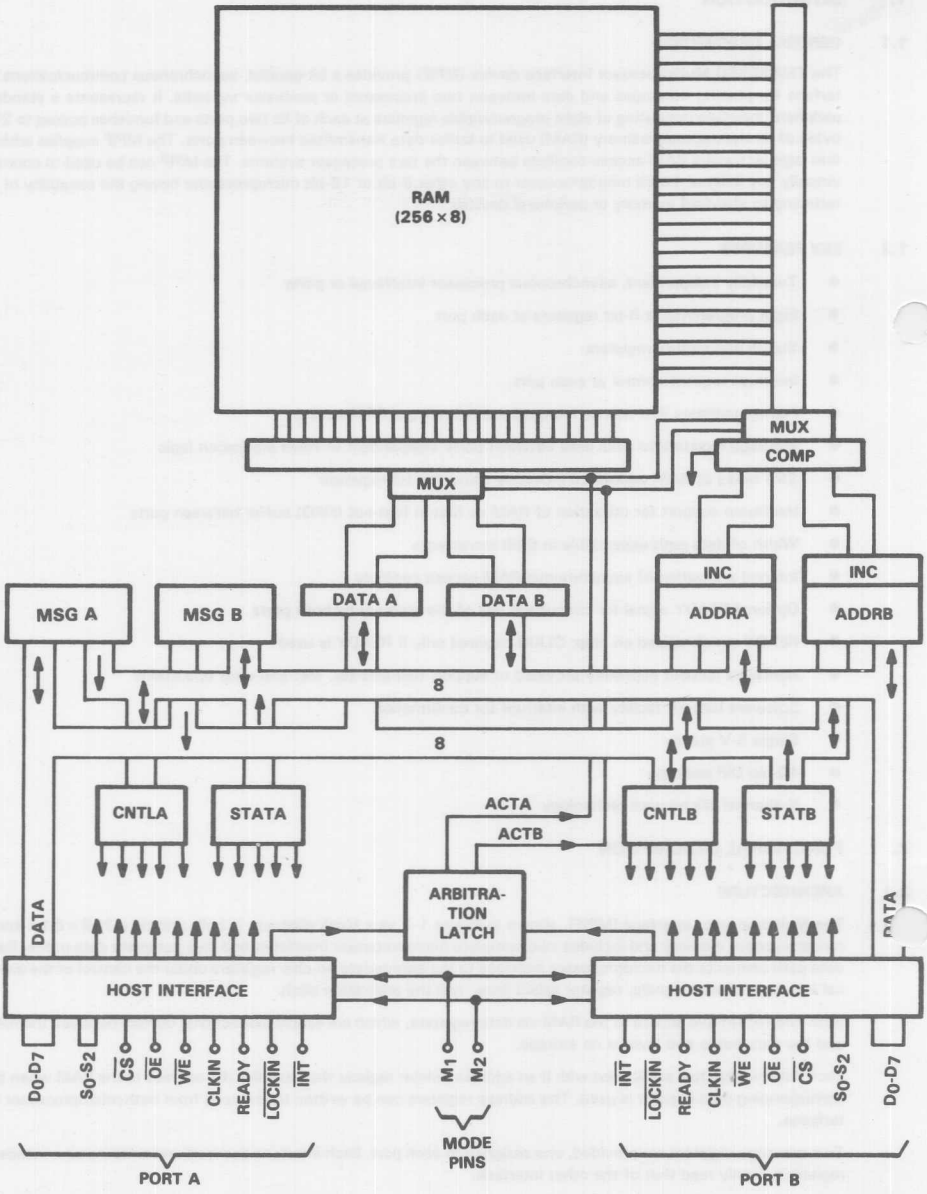


FIGURE 1-1 — MPIF BLOCK DIAGRAM

The status register shows the condition of the various interrupt sources.

The RAM can only be used by one host microprocessor, via its data register, at one time. The two outputs of the arbitration latch, ACTA and ACTB, control access to the RAM. These outputs select the RAM data and address buses from either the DATA A and ADDR A registers or the DATA B and ADDR B registers respectively. ACTA becomes true when the data register of port A is addressed, but only if ACTB is not already true and port B has not asserted a lockout. A corresponding definition applies for ACTB. Hence, the two signals are mutually exclusive, which ensures that both interfaces cannot use the RAM at once. The MPIF provides its host microprocessors with continuous access to all the other registers.

If both interfaces try to gain access to the RAM concurrently, the first to address its data register will exclude the other. The RAM is assigned on a first-come, first-serve basis unless a lockout is in effect (see Section 2.5).

Occasionally, both ports may address their data registers at exactly the same time. This can put the arbitration latch into an indeterminate state for some time. Due to the cross-coupled nature of ACTA and ACTB, the indeterminate state will be unstable. Eventually the conflict will resolve itself, the outcome being essentially random. ACTA and ACTB pass through threshold circuits to ensure that the unstable state is interpreted as an inactive state for both bits.

Each interface can request exclusive use of the RAM using the lockout feature. This is asserted by means of a software-accessible bit or with a dedicated input pin. In situations where both ports assert a lockout, the RAM is assigned on a first-come, first-serve basis.

A memory cycle cannot be allowed to proceed if the port concerned does not succeed in getting access to the RAM or there is uncertainty in the arbitration latch. The problem of sharing the RAM between two ports may be approached in three ways:

1. Ensure in software that both host ports do not try to use the RAM at the same time. Thus any attempt to gain access to the RAM is guaranteed to be successful. This involves the two systems passing messages between themselves regarding their status and intentions, for which the message registers may be used.
2. Use the READY signal provided by the MPIF to put the system into a wait state if it is not successful in gaining access to the RAM or if uncertainty exists in the arbitration latch.
3. Use the software-accessible lockout bit to request exclusive use of the RAM. Wait until this is acknowledged before attempting access to the RAM.

It is possible to use method 2 on one port and 3 on the other port.

2.2 REGISTER DESCRIPTION

The MPIF occupies eight locations in the memory map of each host system. It is so arranged that the registers accessible at the same location of each port serve the same function. The ports of the MPIF are therefore completely identical and can be reversed without software or hardware changes. For the purpose of naming the locations in the memory map, the port under consideration is referred to as the local port and the other is referred to as the remote port. The location and function of each register is shown in Table 2-1.

2.2.1 Data Registers

Each port can read and write its own data register at the two memory locations designated as Data and Data/Increment. If a memory operation is performed to the RAM via the data/increment location, the corresponding address pointer register will be incremented on completion of the memory cycle. This will not happen if the Data location is used.

2.2.2 Address Pointer Registers

Each port can read and write its own address pointer register at the location designated as the Local Address Pointer and can read and write the other port's pointer at the Remote Address Pointer location. This enables each port to determine where in the RAM it will operate by setting up its own address pointer register. Alternatively, the management of the RAM can be under the control of only one port, which sets up both address pointers. Each pointer register will cycle through the value FF_{16} (i.e. increments to 00_{16}). The following limitations apply to the use of these locations:

1. If either address pointer is read while its value is being changed by a write operation from the other port, an erroneous value may be read.

2. A port should not write to its remote address pointer location while there is a possibility that the other port could perform a memory operation to the RAM. This can result in the address pointer being changed during a memory operation and data in the RAM being corrupted.

TABLE 2-1 – MPIF REGISTER MAP

REGISTER SELECT LINES S_0 S_1 S_2	REGISTER FUNCTION	REGISTER SELECTED		READ/WRITE
		PORT A	PORT B	
000	DATA/INCREMENT	DATA A	DATA B	R/W
001	DATA	DATA A	DATA B	R/W
010	MESSAGE IN	MESSAGE B	MESSAGE A	R
011	MESSAGE OUT	MESSAGE A	MESSAGE B	R/W
100	CONTROL	CONTROL A	CONTROL B	R/W
101	LOCAL ADDRESS POINTER	ADDRESS A	ADDRESS B	R/W
110	STATUS	STATUS A	STATUS B	R
111	REMOTE ADDRESS POINTER	ADDRESS B	ADDRESS A	R/W

2.2.3 Message Registers

Each port can read and write its own message register at the location designated as Message Out. In addition, it only reads that of the other port at the Message In location. The message registers are implemented as two 8-bit registers, which can be written to at any time from their corresponding interface. During a write operation, the previous value of the register is held in a latch so that if a read operation occurs concurrently with the write, the previous value of the status register will be read. This means that the hosts may poll their remote status registers at any time without fear of reading an invalid code.

Interrupts are provided to support passing messages (see Section 2.2.5).

2.2.4 Control Registers

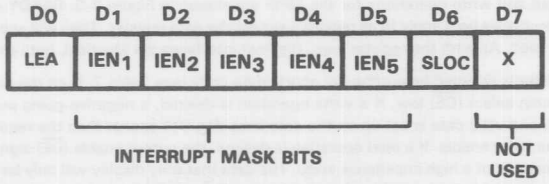
Control registers can be written to and read by their respective hosts at any time. The bit assignment is shown in Figure 2-1.

IEN_1-IEN_5 Interrupt Enable Bits: When set to 1, these allow their respective interrupt status bits to set the INT status bit and pull low the INT line.

LEA Lockout On Equal Addresses Pointer: If this feature is set from either port, it is active for the entire device. When this feature is enabled and the address pointer registers become equal, the port corresponding to the last address pointer register to be loaded from either port or incremented will be locked out of the RAM. The lockout will persist as long as the above condition remains true.

SLOC Software Lockout Bit: This provides a software-accessible means of requesting that the remote port be locked out of the RAM.

All bits of the control register are cleared by the reset function of the mode pins, M_1 and M_2.



LEA – **Lockout on equal address pointers**
IEN_1-IEN_5 – **Enable bits for MI, MO, LPE, RPE, LAK interrupt bits respectively**
SLOC – **Software lockout bit**

FIGURE 2-1 – MPIF CONTROL REGISTER BIT ASSIGNMENT

2.2.5 Status Registers

Each status register is read only and allows its corresponding host to inspect the status of various parameters on-chip. All may cause an interrupt if the appropriate interrupt enable bit is set to a 1. The bit assignment is shown in Figure 2-2.

INT — Interrupt Asserted: An interrupt status bit has been set, and the $\overline{INT}$ line is pulled low.

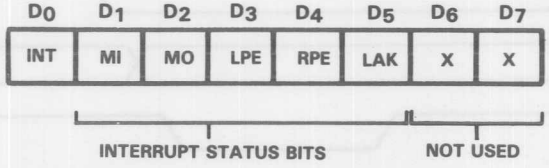
MI — Message In Interrupt: A byte should be read from the Message In register. It is set when the remote port loads its Message Out register and is cleared when the local port reads its Message In register. It is cleared by the reset function.

MO — Message Out Interrupt: The local message register is available for use. It is cleared when a byte is written to the local Message Out register and set when the remote Message In register is read. It is set by the reset function.

LPE — Local Pointer Equal to Remote Pointer: The address pointer registers are equal, and the local pointer was the last one to be loaded from either port or incremented. It remains true as long as the condition persists:

RPE — Remote Pointer Equal to Local Pointer: The address pointer registers are equal, and the remote address pointer was the last one to be loaded from either port or incremented. It remains true as long as the condition persists.

LAK — Lockout Acknowledge: This is set following the assertion of SLOC by the local port when the lockout of the remote port from the RAM becomes effective. It is cleared when SLOC is cleared.



INT – **Interrupt occurred**
MI – **Message in**
MO – **Message out**
LPE – **Local address pointer equal to remote address pointer**
RPE – **Remote address pointer equal to local address pointer**
LAK – **Lockout acknowledge**

FIGURE 2-2 – MPIF STATUS REGISTER BIT ASSIGNMENT

2.3 HOST INTERFACES

The simplest read and write operations for the MPIF are shown in Figure 2-3. READY and CLKIN are not shown since the memory cycles here apply to all registers except the data register. They will apply to the data register only if READY is not used. As with the register map, the host interfaces are identical, both electrically and functionally.

The desired register is selected by putting the appropriate code (see Table 2-1) on the register select lines (S_0-S_2) and by putting chip select ($\overline{CS}$) low. If a write operation is desired, a negative-going pulse is applied to the write enable pin ($\overline{WE}$), and valid data is set up on the data lines (D_0-D_7) sooner than the required setup time before the rising edge of the write enable. If a read operation is desired, the output enable ($\overline{OE}$) signal is set low, which brings the MPIF data lines out of a high impedance state. The data that they display will only be valid after the appropriate access time has elapsed from the register being selected. The required setup times, access times, etc. are given in Section 4.4.

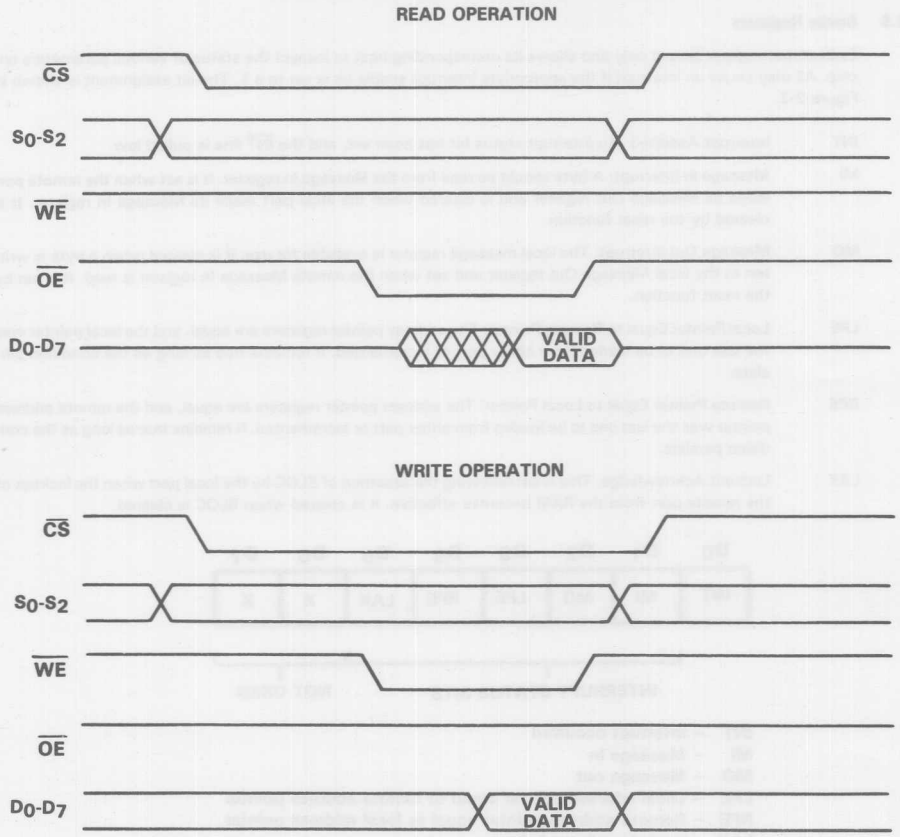


FIGURE 2-3 — MPIF READ AND WRITE OPERATIONS

2.4 READY AND CLKIN

Although the MPIF host interfaces can function without the READY and CLKIN signals, both signals are required if concurrent access to the RAM is desired by both host systems. Under these conditions, the selection of one interface or the other on to the RAM is done by the arbitration latch. The host interface logic is responsible for putting into a wait state the host which is unsuccessful in gaining access.

When a host system addresses the data register of the MPIF (chip select low and the appropriate code on the register select lines), READY is immediately set low regardless of whether or not access is actually gained to the RAM. READY will then stay low, and the interface will remain in a wait state until any uncertainty in the arbitration latch has resolved itself and access has been clearly gained.

Since the majority of systems will not accept an asynchronous READY signal, synchronization is provided on the MPIF. The falling edge of READY is generated by the CPU addressing its data register, so it is already synchronous. The rising edge, however, is not and must be synchronized to the system clock. CLKIN is provided for this purpose alone.

Each output of the arbitration latch is monitored by a threshold detector, which tests for a level in excess of the metastable level. ACTA or ACTB reaching this level indicates any conflict has resolved itself, and the corresponding port has gained access to the RAM. During the CLKIN high period, the output of the threshold detector is sampled as shown in Figure 2-4. When CLKIN is low, the feedback is applied to consolidate the sampled value so that any indeterminate sample will go to a valid 1 or 0 level. If a 1 is detected indicating that the memory cycle can proceed, then READY is set high on the next rising edge of CLKIN.

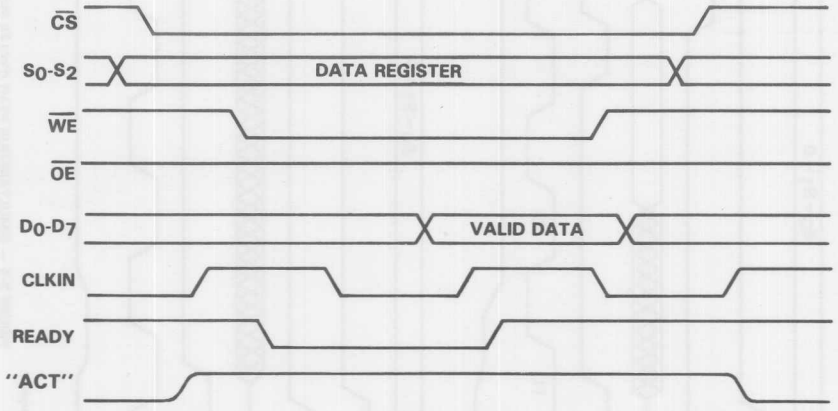
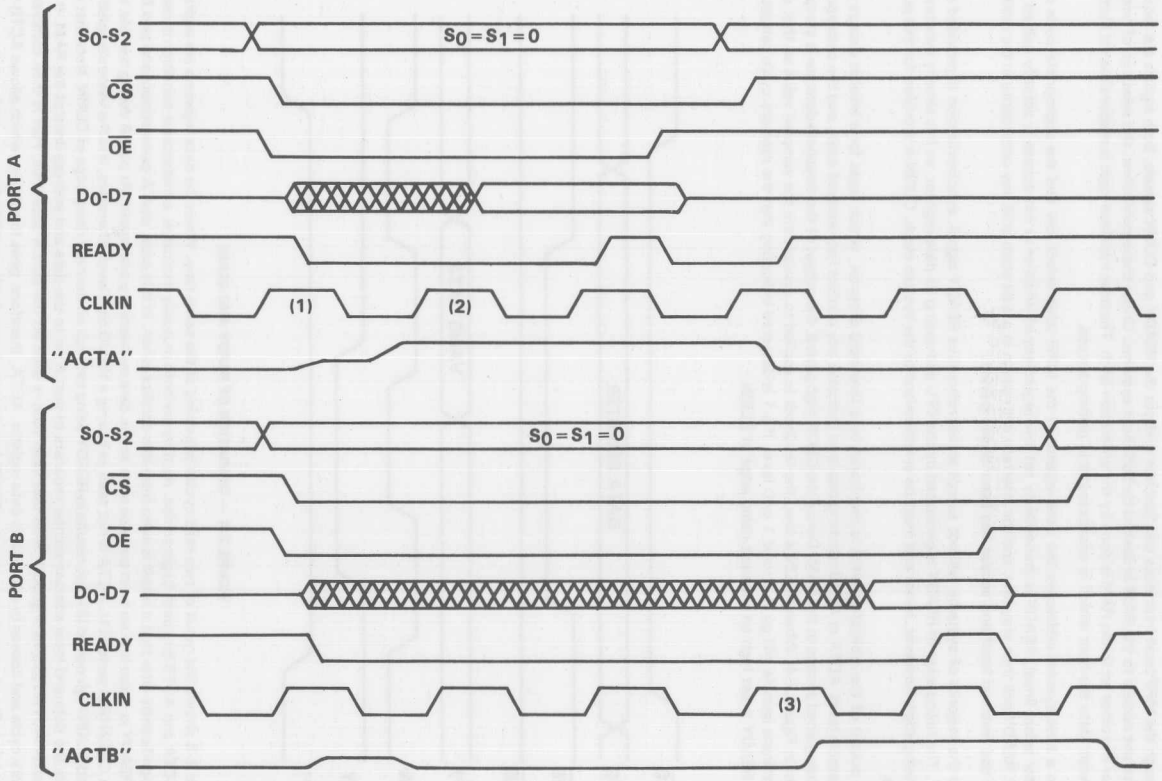


FIGURE 2-4 — OPERATION OF READY AND CLKIN

Figure 2-5 shows the result of two read cycles beginning at the same time. When the data registers are addressed, the ACTA and ACTB bits both begin to rise. As they are both mutually exclusive, a metastable condition is reached. It is significantly later that a result is seen from this conflict when, in this case, port A gains access and port B does not. READY is taken low on both ports as soon as the data registers are addressed. At port A during the first subsequent CLKIN high period (1), ACTA is not seen as having a VALID high level. Therefore, it is the value sampled in the second CLKIN high period (2) that results in READY being set high on the next rising edge of CLKIN. Note that ACTA selects the data and local address pointer registers of port A on to the data and address lines of the RAM. It is not until after ACTA reaches a good 1 level that valid data is seen on the port A data lines. Port A then completes its memory cycle and ceases to address its data register. ACTA, therefore, goes low again, which allows ACTB to rise to a 1 level. This fact is detected in the CLKIN high period of port B (3), and READY goes high on the next rising edge. Up to this point, the data lines of port B have been displaying invalid data. This becomes valid after ACTB reaches a good 1 level. The memory cycle at port B can then be completed.



NOTE: ACTA and ACTB are internal signals.

FIGURE 2-5 — SIMULTANEOUS READ CYCLES FROM BOTH PORTS

2.5 LOCKOUT CAPABILITY

Both ports have a lockout feature that can be asserted by either one of two means: (1) by putting a low level on the $\overline{\text{LOCKIN}}$ input of the host interface, or (2) by writing a 1 to the SLOC bit of the control register. If a lockout is asserted by the local port, then the ACT bit of the remote port is held low. Thus if the remote port addresses its data register, it will not get access to the RAM. If CLKIN and READY are used on the remote port, it will enter a wait state until the lockout is removed.

The assertion of a lockout will not guarantee immediate exclusive use of the RAM. A lockout asserted by the local port will only become effective after any memory operation to the RAM by the remote port has been completed. It will also not be effective until any lockout asserted by the remote port has been cleared. Lockouts, therefore, are mutually exclusive in a similar way to ACTA and ACTB, and concurrent lockout requests from both ports are assigned on a first-come, first-serve basis.

Figure 2-6 shows an example of the lockout facility being used to implement an indivisible read-modify-write operation. Port A performs the read and write operations with a lockout asserted between them by means of the $\overline{\text{LOCKIN}}$ input. At the time when port A tries to do a read from the RAM (1), there is already a read cycle in progress on port B. Port A, therefore, enters a wait state until this operation is complete even though the $\overline{\text{LOCKIN}}$ input is asserted. When the read cycle at port B is complete, the memory operation at port A can proceed and, in addition, the lockout of port B becomes effective. Consequently, when the next memory operation is initiated to port B (2), it enters a wait state even though there is no activity on port A. Port A then enters a write cycle (3), during which the $\overline{\text{LOCKIN}}$ is removed. When the write cycle ends and ACTA goes low (4), ACTB can rise in the absence of the lockout and the memory operation can also be completed here. Therefore, the read and write operations at port A cannot be interfered with from port B. $\overline{\text{LOCKIN}}$ would be derived from a multiprocessor interlock-type signal in host system A. The $\overline{\text{LOCKIN}}$ signal has no effect unless $\overline{\text{CS}}$ is active low and $S0 = S, = 0$ (i.e., RAM access).

In a system where the user does not wish to use the READY and CLKIN signals on a particular port, the SLOC bit can be used to guarantee that access is gained to the RAM. The LAK interrupt status bit will be set in response to SLOC as soon as the lockout becomes effective. Thus LAK will not be set until any current memory cycle to the RAM from the remote port has been completed and any lockout that the remote port may have asserted has been cleared. After this, the local port has exclusive use of the RAM until it clears SLOC.

2.6 ADDRESS POINTER EQUAL INTERRUPTS AND LOCKOUT

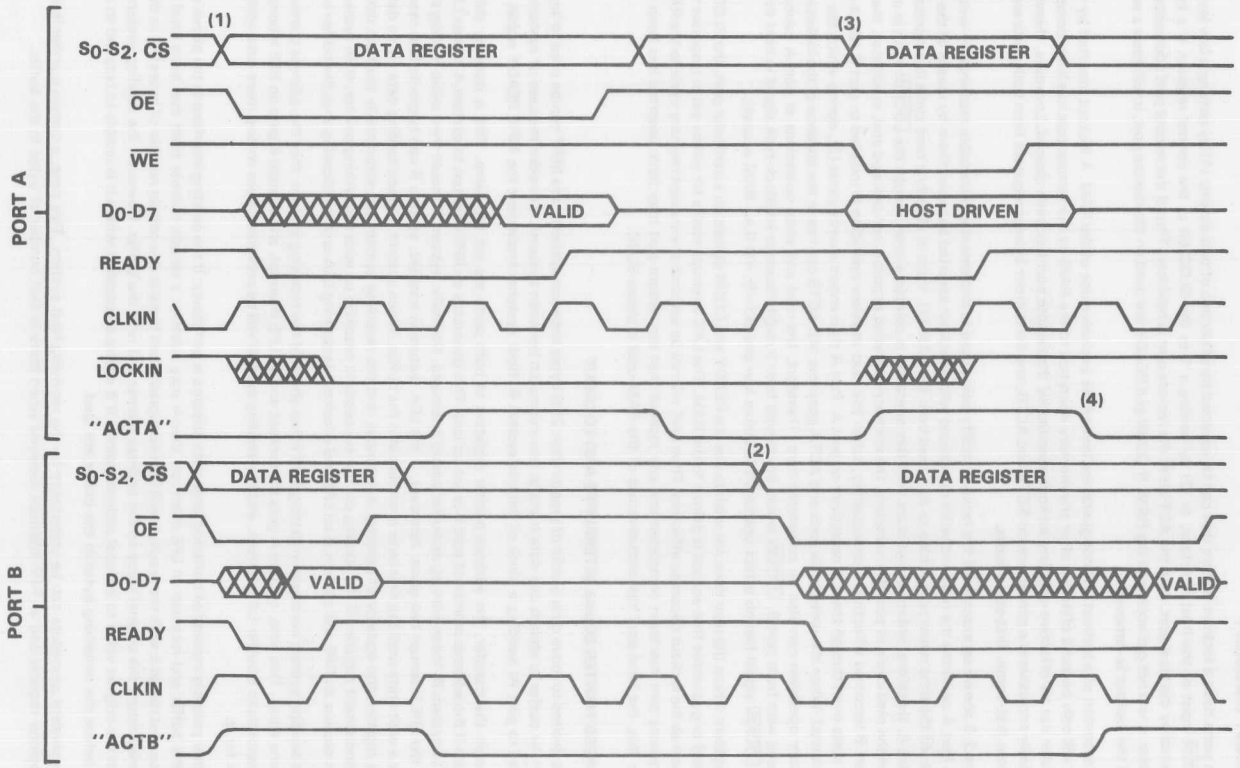
If it is desired to move data blocks of greater than 256 bytes between systems, the MPIF can be used to implement a circular buffer to absorb any data transfer rate mismatch between systems. Consider the case of system A (connected to port A) sending a block of data to system B. Both systems implement the MPIF READY signal.

To begin the transfer, the address pointer registers of both ports are set the same. This is done by either host system if the address pointer of port B is set up last. This generates an RPE interrupt to system A and an LPE interrupt to system B, these serving as buffer empty interrupts. Normally, system B should then avoid reading the RAM until the LPE interrupt has gone. However, if the LEA feature is enabled, system B can begin its first read. It will enter a wait state until the first byte is written into the buffer. When system A starts loading data via its data/increment register, the equality of pointers is removed, and the receiving system can sequentially read the data via its data/increment register. If the reading pointer succeeds in catching up with the writing pointer, then system A will again receive an RPE and system B an LPE and a lockout (assuming LEA is set) indicating that the buffer is empty.

If the sending system succeeds in getting 256 bytes ahead of the receiving system, then the address pointers again become equal. This time, system A gets the lockout and an LPE interrupt, and system B gets an RPE interrupt. This corresponds to a buffer full interrupt, and the sending system will be prevented from writing more data until there is room for it.

Another possible method of buffering large data streams is as follows: If the sending interface in the above example fills the buffer and receives an LPE interrupt, then it may subtract a certain number from that in its local address pointer and reload it with the result. It will be interrupted when there is this certain number of bytes left in the buffer and may return the original value to the address pointer and refill the buffer. However, the sending system should return the original value to its local address pointer if it enters a condition where it is unable to respond to an interrupt before the remaining bytes in the buffer are read.

An equivalent procedure can be undertaken by the receiving host system. This time, a certain number is added to the pointer register and an RPE interrupt received when there is that number of bytes in the buffer.



NOTE: ACTA and ACTB are internal signals.

FIGURE 2-6 — TYPICAL READ-MODIFY-WRITE OPERATION USING $\overline{\text{LOCKIN}}$

2.7 MODE PINS

The mode pins, M_1 and M_2, are used to reset the MPIF and to enable several MPIFs to be used in parallel on memory buses of greater than 8 bits. There are four modes encoded on these pins: reset ($M_1 = M_2 = 0$), standalone ($M_1 = M_2 = 1$), master ($M_1 = 0$, $M_2 = 1$), and slave ($M_1 = 1$, $M_2 = 0$). Schmitt triggers are provided on both inputs to permit the use of a resistor and capacitor arrangement to implement reset.

2.7.1 Reset ($M_1 = M_2 = 0$)

The reset function establishes the following conditions on-chip:

1. All bits of the control register cleared
2. The MI interrupt status bit cleared
3. The MO interrupt status bit set
4. The data lines (D_0-D_7) of the host interfaces held in a high impedance state
5. The READY output of each port held in a high impedance state.

The other three combinations of the mode pins are operating modes.

2.7.2 Standalone Mode ($M_1 = M_2 = 1$)

The standalone mode is the operating mode of a single MPIF. To implement reset with this mode of operation, both M_1 and M_2 should be connected to an active-low system $\overline{\text{RESET}}$ signal.

2.7.3 Master ($M_1 = 0$, $M_2 = 1$) and Slave ($M_1 = 1$, $M_2 = 0$) Modes

The master and slave modes are included to avoid the possibility of problems occurring in multiple MPIF arrangements. During simultaneous attempts at getting access to the RAM by both ports, it is possible for the arbitration latches of different devices in standalone mode to fall in opposite directions with consequent system malfunction. Master and slave modes allow the arbitration latch in only one MPIF to decide which port should have access. This decision is then passed on to the remainder. Figure 2-7 shows an example of a multiple MPIF system.

To implement the reset function on the master device, M_2 should be connected to the system RESET signal, and M_1 should be grounded. In master mode operation, the timing of the READY line of each port is changed to provide an unclocked, active-high indication of when that port has gained access to the RAM. A CLKIN input is, therefore, not required by the master device.

On the slave devices, M_1 is connected to the system $\overline{\text{RESET}}$ line, and M_2 is grounded. In this mode, the $\overline{\text{LOCKIN}}$ signals of each port become enable inputs, which are connected directly to the modified READY outputs of the corresponding port of the master device. The slave has no arbitration and responds to a high level on the READY output of the master by granting access to the appropriate port immediately. At this time, it also begins the procedure of releasing its own READY line which is synchronized in the same way as on the standalone device. Hence, a CLKIN must be supplied to the slave devices. In dual MPIF arrangements, the READY outputs of the slave may be taken directly to the READY input of the host systems. With more than one slave, the READY outputs of each port should be ANDed together to ensure that all MPIFs give access to the port before READY is released.

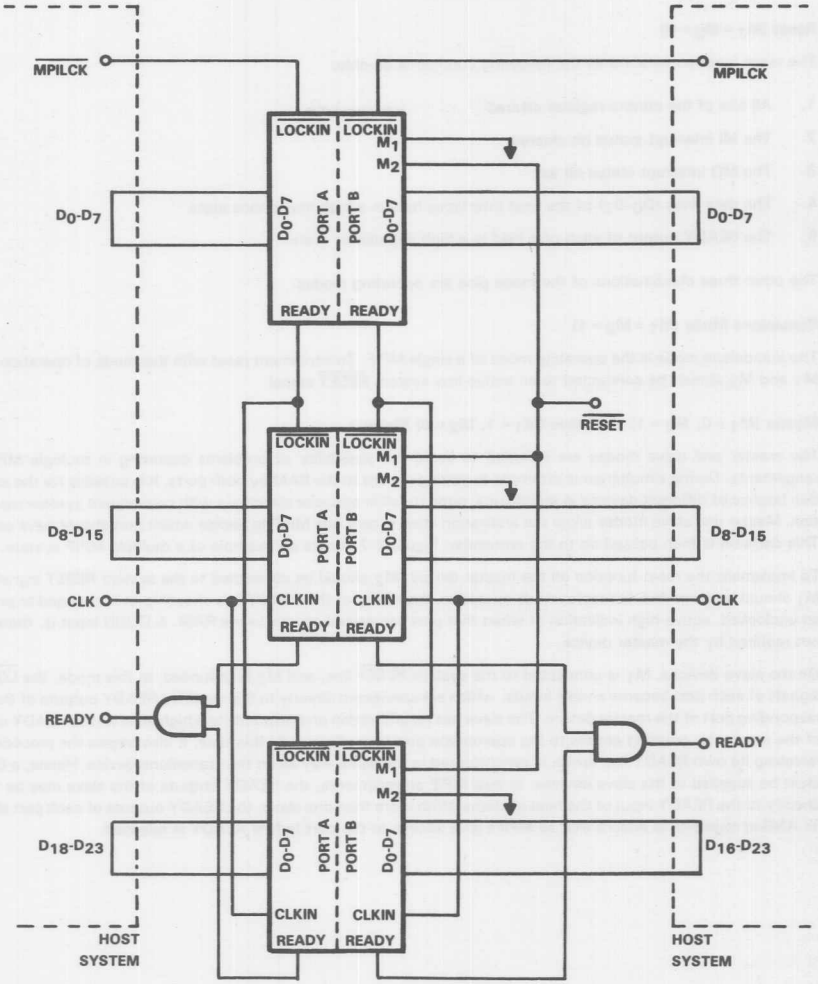


FIGURE 2-7 — MULTIPLE MPIF CONFIGURATION

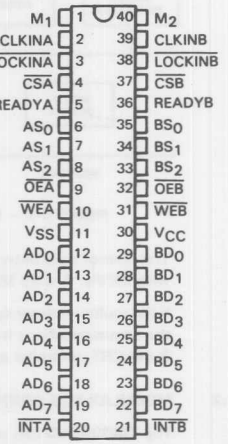
2.8 PIN DESCRIPTION

Table 2-2 defines the TMS99650 pin assignments and describes the function of each pin.

TABLE 2-2 – PIN ASSIGNMENT AND FUNCTION

SYMBOL	PIN NUMBER		TYPE I/O	DESCRIPTION
	PORT A	PORT B		
D_0 (MSB)	12	29	I/O	DATA BUS: Provides for bidirectional data transfer between the MPIF port and the host system.
D_1	13	28	I/O	
D_2	14	27	I/O	
D_3	15	26	I/O	
D_4	16	25	I/O	
D_5	17	24	I/O	
D_6	18	23	I/O	
D_7 (LSB)	19	22	I/O	
S_0	6	35	I	REGISTER SELECT LINES: Indicate to the MPIF which internal register is accessed by the host system.
S_1	7	34	I	
S_2	8	33	I	
$\overline{CS}$	4	37	I	CHIP SELECT: Indicates that the host system requires access to one of the MPIF internal registers.
$\overline{WE}$	10	31	I	WRITE ENABLE: Indicates that the host is performing a write operation.
$\overline{OE}$	9	32	I	OUT ENABLE: Indicates that the host system is performing a read operation.
READY	5	36	O	READY: Indicates to the host system that the memory operation in progress may be completed.
CLKIN	2	39	I	CLOCK-IN: Allows READY to be presented synchronously to the host system.
$\overline{LOCKIN}$	3	38	I	LOCKOUT IN: Indicates to the MPIF that the opposite port should be denied access to the RAM.
$\overline{INT}$	20	21	O(o/d)*	INTERRUPT: Indicates to the host system that it should branch to a service routine.
M_1	1			MODE PINS: Reset the MPIF and establish whether it is to work in master, slave, or standalone mode.
M_2	40			
V_{CC}	30			POWER SUPPLY
V_{SS}	11			GROUND REFERENCE

* o/d = open drain output: all others are push/pull outputs.



3. APPLICATIONS

Examples of TMS99650 applications are discussed in the following paragraphs.

3.1 PARTITIONING OF SYSTEM FUNCTIONS

Using the TMS99650, functions can be partitioned among as many processors as required to achieve a desired level of performance. Figure 3-1 shows a system with three processors and two MPIFs.

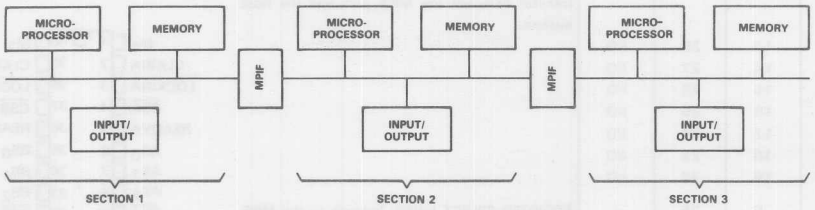


FIGURE 3-1 — PARTITIONING OF SYSTEM FUNCTIONS AMONG THREE MICROPROCESSORS

The flow of data between sections 1 and 2, and sections 2 and 3 is high enough to warrant linking them together with MPIFs. A third MPIF is not required if little or no data is exchanged between sections 1 and 3.

If the multiprocessor system shown in Figure 3-1 resides on a controller board for an intelligent terminal, section 1 is the communications front-end processor, section 2 the keyboard monitor and high-level user interface, and section 3 the CRT controller and graphics processor.

3.2 MODULAR-BUS CONFIGURATION

The TMS99650 can serve as the slave interface through which an intelligent peripheral controller connects to a system bus controlled by a host processor. In Figure 3-2, the host processor controls the system bus, which is also connected to a main memory and two intelligent function modules.

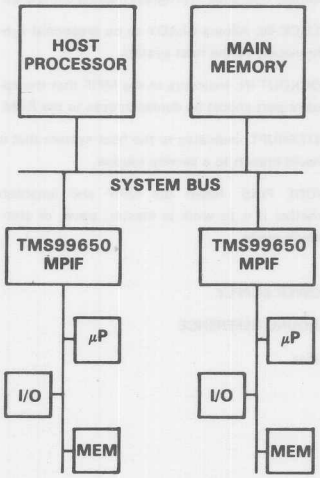


FIGURE 3-2 — FUNCTION MODULES CONNECTED TO SYSTEM BUS

Each function module contains an MPIF, processor, memory, and I/O capability. These and other similar modules are easily connected to or removed from the bus to configure the system as required. The host presents a master interface capable of actively initiating bus transfers. The other modules, memory and MPIF, present passive or slave interfaces to the bus. These modules can transmit or receive data only under the direct influence of the master. Whereas a master interface tends to be relatively complex to provide these capabilities, the MPIF interfaces to the bus as easily as a memory or any other device with a slave interface.

A direct memory access (DMA) controller could be added to the system bus to help the host processor transfer blocks of data between the main memory and the function modules. Each word transferred via DMA consumes two bus cycles. For example, in moving data from a function module to the main memory, the DMA controller first reads each word from the module, and then it writes the word to memory.

3.3 INTERFACING AN 8-BIT TO A 16-BIT MICROPROCESSOR SYSTEM

In the example shown in Figure 3-3, an interface is constructed of two TMS99650 MPIFs configured in standalone mode. The two READY signals at ports A and B are gated together to form composite READY signals to the 8- and 16-bit systems. Clocks synchronize the READY signals.

The 16-bit system on the right side of Figure 3-3 accesses the two MPIFs in parallel; the upper MPIF inputs or outputs data on lines D0-D7, and the lower MPIF inputs or outputs data on lines D8-D15 of the 16-bit system. On the left side, the 8-bit system accesses each MPIF separately; both MPIFs input or output data on the same eight data lines, D0-D7, of the 8-bit system.

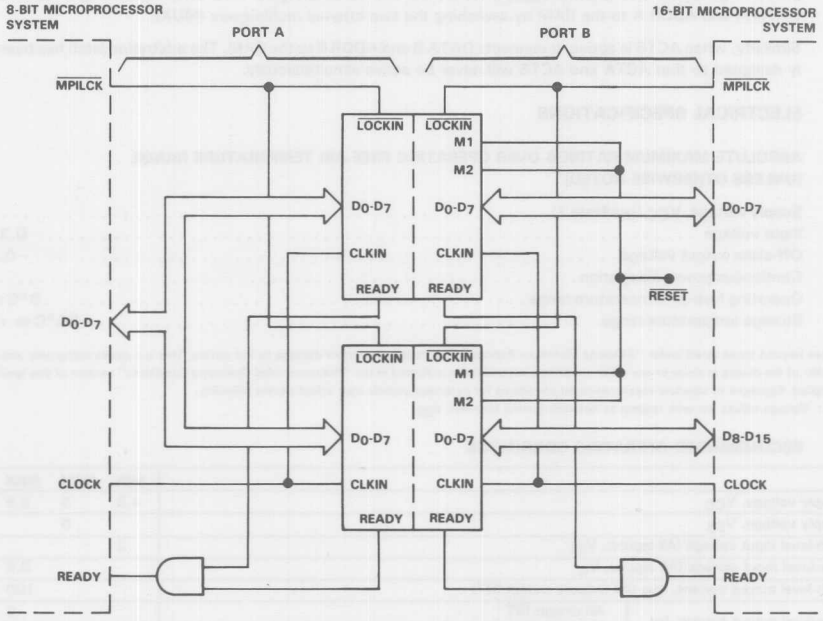


FIGURE 3-3 — INTERFACING AN 8-BIT TO A 16-BIT MICROPROCESSOR SYSTEM

To avoid bus deadlock when asserting $\overline{\text{LOCKIN}}$, the 8-bit system should attempt to lock out the 16-bit system from either one MPIF or the other, but never both simultaneously. In applications where a configuration is needed in which either system can safely obtain exclusive access to both MPIFs simultaneously, one MPIF must be configured in master mode and the other in slave mode, as described earlier.

master and slave modes, the operation of these mechanisms is modified.

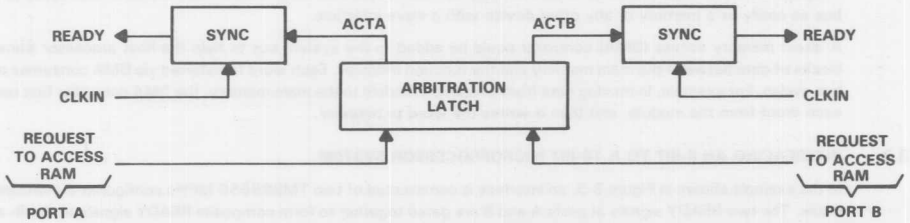


FIGURE 3-4 — ARBITRATION AND SYNCHRONIZATION

The MPIF contains all arbitration and synchronization logic needed to maintain the integrity of the interface between two independent and asynchronous processors. As indicated in Figure 3-4, the arbitration latch is responsible for granting RAM access to one port or the other. The arbitration latch receives requests for RAM access from either port and generates two internal signals, ACTA and ACTB. (See Figure 1-1). When active high, ACTA connects DATA A and ADDR A to the RAM by switching the two internal multiplexers (MUX).

Similarly, when ACTB is active, it connects DATA B and ADDR B to the RAM. The arbitration latch has been carefully designed so that ACTA and ACTB will never be active simultaneously.

4. ELECTRICAL SPECIFICATIONS

4.1 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)*

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . −0.3 to 20 V
Off-state output voltage . . . . . −0.3 to 7 V
Continuous power dissipation . . . . . 1 W
Operating free-air temperature range . . . . . 0 °C to 70 °C
Storage temperature range . . . . . −55 °C to +150 °C

* Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: Voltage values are with respect to network ground terminal, V_{SS}.

4.2 RECOMMENDED OPERATING CONDITIONS

		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}		4.5	5	5.5	V
Supply voltage, V_{SS}			0		V
High-level input voltage (All inputs), V_{IH}		2			V
Low-level input voltage (All inputs), V_{IL}				0.8	V
High-level output current, I_{OH} (All outputs except $\overline{INT}$)				100	μA
Low-level output current, I_{OL}	All except $\overline{INT}$			2	mA
	$\overline{INT}$ output only			2.5	
Operating free-air temperature, T_A		0		70	°C

4.3 ELECTRICAL CHARACTERISTICS OVER RECOMMENDED FREE-AIR TEMPERATURE RANGE

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{OH}	High-level output voltage	V_{CC} = min,	I_{OH} = max	2.4			V
V_{OL}	Low-level output voltage	V_{CC} = min,	I_{OL} = max			0.4	V
I_O	Off-state (high-impedance state) output current	V_{CC} = max	V_O = 2.4 V			±20	μA
			V_O = 0.4 V			±20	
I_I	Input current	V_I = V_{SS} to V_{CC}				±50	μA
I_{OS}	Short-circuit output current§	V_{CC} = max					μA
I_{CC}	Supply current	V_{CC} = max				200	μA
C_i	Input capacitance (except data bus)	f = 1 MHz, all other pins at 0 V			15		pF
C_{DB}	Data bus capacitance				25		pF
C_o	Output capacitance (except data bus)				10		pF

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.
§ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

4.4 TIMING REQUIREMENTS AND CHARACTERISTICS

PARAMETER		CONDITION	SEE NOTE	SEE FIGURE	MIN	TYP	MAX	UNIT
t_{SU1}	Register select setup time for write	Data register $S_0 = S_1 = 0$		4-1	205			ns
		Otherwise		4-1	150			
t_{SU2}	Chip select setup time for write	Data register $S_0 = S_1 = 0$		4-1	195			ns
		Otherwise		4-1	140			
t_{SU3}	Data setup time for write	Data register $S_0 = S_1 = 0$		4-1	150			ns
		Otherwise		4-1	50			
t_{WL2}	Write enable low pulse width	Data register $S_0 = S_1 = 0$		4-1	200			ns
		Otherwise		4-1	70			
t_{H1}	Data hold time for write			4-1	0			ns
t_{H2}	Chip select hold time for write			4-1	0			ns
t_{H3}	Register select hold time for write			4-1	0			ns
t_{H4}	Hold time of write enable after CLKIN rising edge during concurrent RAM accesses	Data register	7	4-4	140			ns

NOTES:
1. Figure 4-6 shows the load circuit used to measure the timing characteristics of output and I/O pins. A value of C_L = 100 pF is used except where otherwise stated.
2. These times only apply when the port in question gets immediate access to the RAM. Otherwise, the access time is determined by t_{AC4}.
3. Only one of the times t_{H5} or t_{H6} need be satisfied. These specify the maximum length of a RAM read operation after access has been gained.
4. These setup times need to be met if READY is to be set high on the next rising edge of CLKIN. If these setup times are not met, then READY will not be released until one CLKIN cycle later (provided a memory access or lockout is not in effect at the remote port).
5. This setup time is required if $\overline{\text{LOCKIN}}$ is to be effective immediately after the termination of the memory access to the RAM. The memory access may be terminated either by $\overline{\text{CS}}$ going high or the address lines changing.
6. This is the delay of the interrupt line from the termination of the memory cycle that causes it. The cycle is terminated when $\overline{\text{CS}}$ goes high or when either $\overline{\text{OE}}$ or $\overline{\text{WE}}$ goes high.
7. These parameters describe the access time and required WE hold times when access to the RAM is not immediately achieved and the host system enters more than one wait state. The parameters are measured from the falling edge of CLKIN on which the corresponding ACT bit is first sampled as being high. This sampled value indicates that access to the RAM has started and results in READY are being released on the next rising edge of CLKIN.

TIMING REQUIREMENTS AND CHARACTERISTICS (Concluded)

		CONDITION	SEE NOTE	SEE FIGURE	MIN	TYP	MAX	UNIT
t_{AC1}	Access time from register select for read	Data register $S_0 = S_1 = 0$	1,2	4-2			185	ns
		Otherwise	1	4-2		85		
t_{AC2}	Access time from chip select for read	Data register	1,2	4-2			175	ns
		Otherwise	1	4-2		75		
t_{AC3}	Access time from output enable for read	Data register $S_0 = S_1 = 0$	1,2	4-2			100	ns
		Otherwise	1	4-2		40		
t_{P1}	Chip select to data bus Hi-Z		1	4-2		40		ns
t_{P2}	Output enable to data bus Hi-Z		1	4-2			30	ns
t_{AC4}	Access time from CLKIN low during concurrent RAM accesses	Data register	1,7	4-4			120	ns
		Data register C_L = 25 pF	1,7	4-4			100	
t_{H5}	Chip select hold time after valid data	Data register	3	4-2			2	μs
t_{H6}	Output enable hold time after valid data	Data register	3	4-2			2	μs
t_{P3}	Chip select to ready low	Data register	1	4-3		40		ns
t_{P4}	Register select to ready low	Data register	1	4-3		50		ns
t_{P5}	CLKIN to ready high	Data register	1	4-3		40		ns
		Data register C_L = 25 pF	1	4-3			20	
t_{SU4}	Chip select setup time to CLKIN	Data register	4	4-3	50			ns
t_{SU5}	Register select setup to CLKIN	Data register	4	4-3	60			ns
t_{SU6}	$\overline{\text{LOCKIN}}$ setup to end of access	Data register	5	4-3	100			ns
t_{WL1}	CLKIN low pulse width			4-3	60			ns
t_{WH1}	CLKIN high pulse width			4-3	45			ns
t_{P6}, t_{P7}	End of memory cycle to interrupt	4-5	1,6	4-5			200	ns
t_r	CLKIN rise time					5	15	ns
t_f	CLKIN fall time					5	15	ns

NOTES:
1. Figure 4-6 shows the load circuit used to measure the timing characteristics of output and I/O pins. A value of C_L = 100 pF is used except where otherwise stated.
2. These times only apply when the port in question gets immediate access to the RAM. Otherwise, the access time is determined by t_{AC4}.
3. Only one of the times t_{H5} or t_{H6} need be satisfied. These specify the maximum length of a RAM read operation after access has been gained.
4. These setup times need to be met if READY is to be set high on the next rising edge of CLKIN. If these setup times are not met, then READY will not be released until one CLKIN cycle later (provided a memory access or lockout is not in effect at the remote port).
5. This setup time is required if $\overline{\text{LOCKIN}}$ is to be effective immediately after the termination of the memory access to the RAM. The memory access may be terminated either by $\overline{\text{CS}}$ going high or the address lines changing.
6. This is the delay of the interrupt line from the termination of the memory cycle that causes it. The cycle is terminated when $\overline{\text{CS}}$ goes high or when either $\overline{\text{OE}}$ or $\overline{\text{WE}}$ goes high.
7. These parameters describe the access time and required $\overline{\text{WE}}$ hold times when access to the RAM is not immediately achieved and the host system enters more than one wait state. The parameters are measured from the falling edge of CLKIN on which the corresponding ACT bit is first sampled as being high. This sampled value indicates that access to the RAM has started and results in READY are being released on the next rising edge of CLKIN.

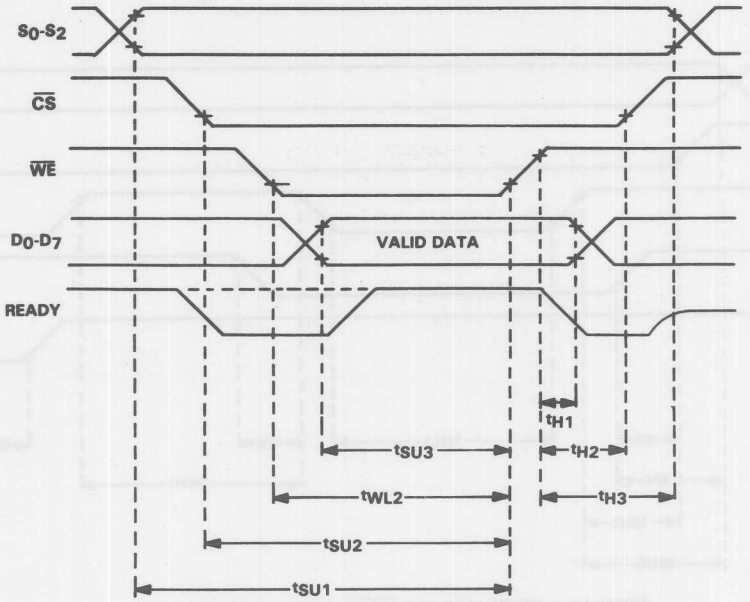


FIGURE 4-1 — WRITE CYCLE TIMING CHARACTERISTICS

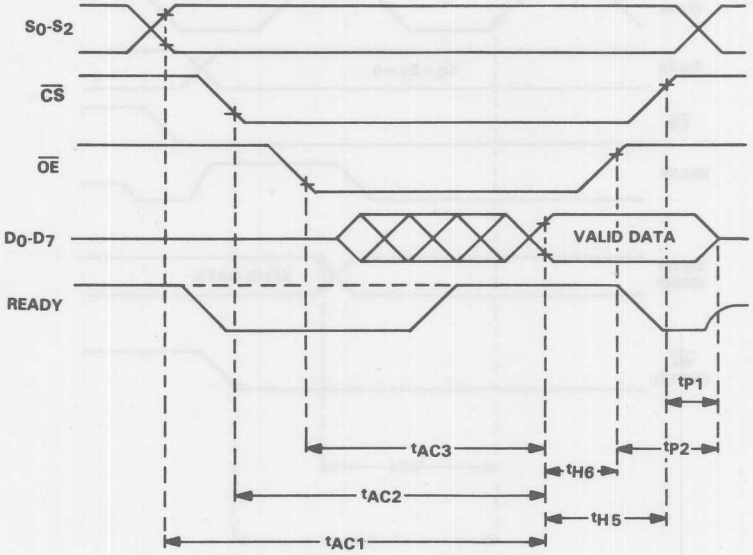


FIGURE 4-2 — READ CYCLE TIMING CHARACTERISTICS

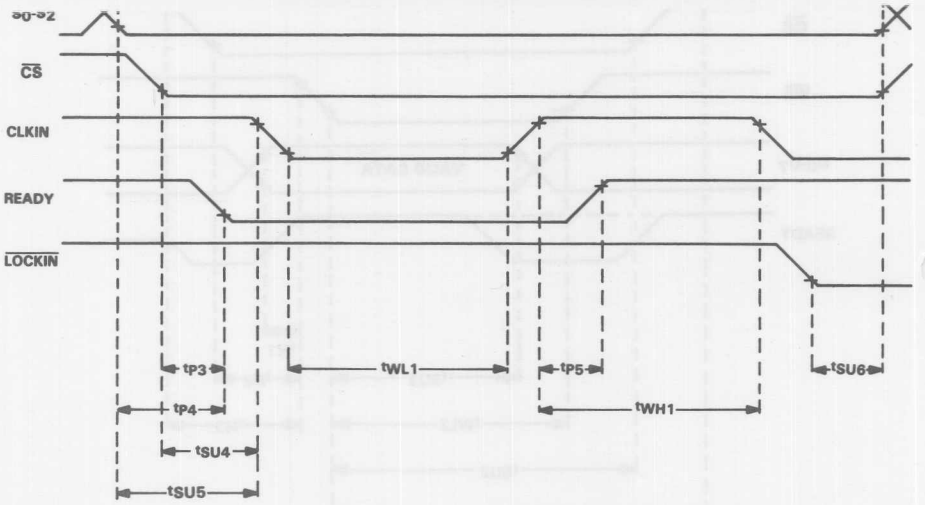


FIGURE 4-3 — READY, CLKIN AND LOCKIN TIMING CHARACTERISTICS

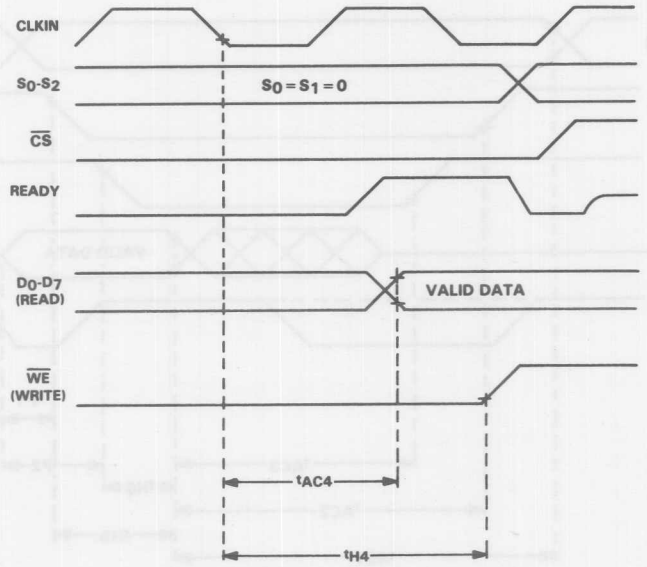


FIGURE 4-4 — READ AND WRITE CYCLE TIMING DURING RAM ACCESSES

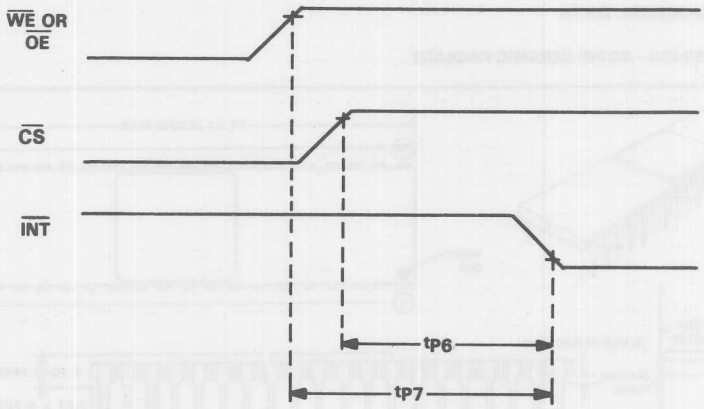
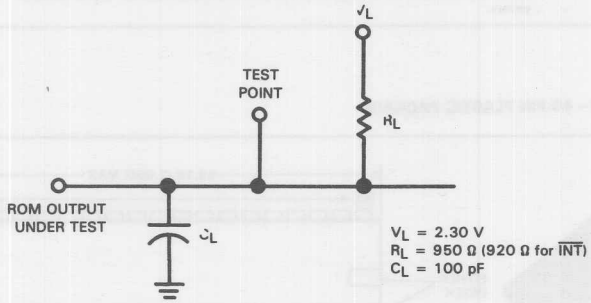


FIGURE 4-5 — INTERRUPT TIMING CHARACTERISTICS

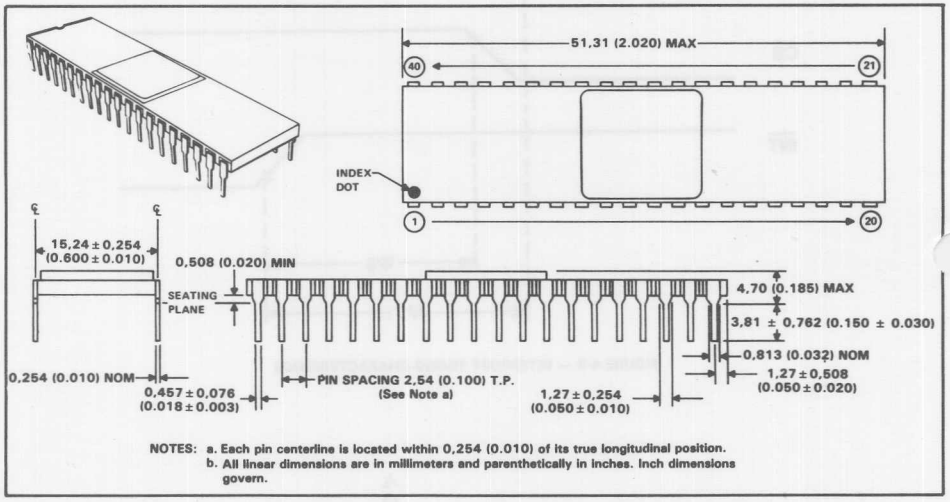


All diodes are IN916 or IN3064.

FIGURE 4-6 — TEST LOAD CIRCUIT

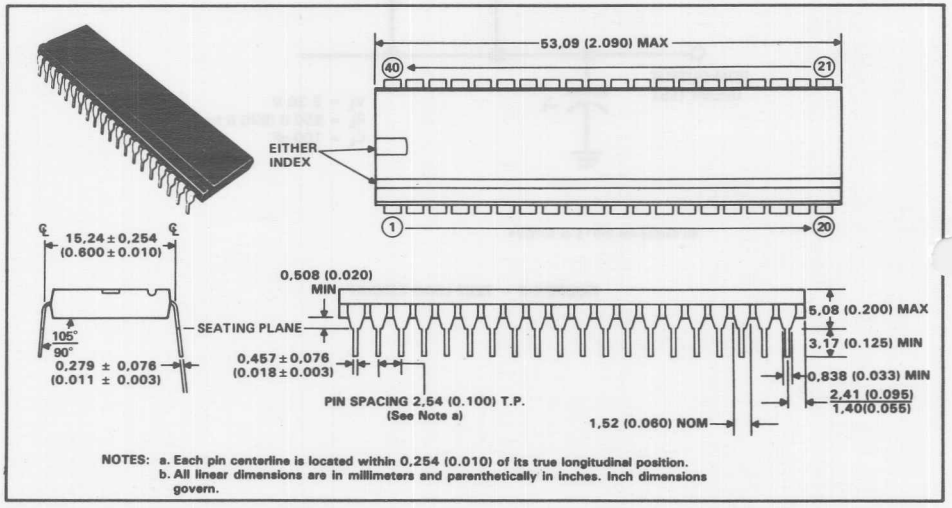
5. MECHANICAL DATA

5.1 TMS99650 – 40-PIN CERAMIC PACKAGE



NOTES: a. Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.
b. All linear dimensions are in millimeters and parenthetically in inches. Inch dimensions govern.

5.2 TMS99650 – 40-PIN PLASTIC PACKAGE



NOTES: a. Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.
b. All linear dimensions are in millimeters and parenthetically in inches. Inch dimensions govern.